

XporY Trader API (P2P) — Documento Técnico

Escopo: Fluxo de compra WL↔WL entre contas trader, garantindo veracidade de estoque/preço no exportador.

1. Objetivos

- Confirmar compra no **WL exportador** antes de capturar a venda local no **WL importador**.
- Garantir **idempotência, baixa latência e resiliência** (CB, retry, timeouts).

2. Arquitetura Lógica

- **Endpoint POST /trader/purchase** (exportador)
- **Orquestração** no **POST /checkout/imported-offer/{id}** (importador)
- **Idempotency Store** (TTL ≥ 24h)
- **Circuit Breaker** por par WL↔WL

3. Contratos — Descrição dos Endpoints

- **POST /trader/purchase** (exportador): payload mínimo `{originOfferId, quantity, requesterWLIId, requesterTraderId, clientBuyerMasked}`; headers `Idempotency-Key, Authorization`; mTLS obrigatório. Respostas: `CONFIRMED|REJECTED` com `tradeId, finalQty, unitPrice`.
- **POST /checkout/imported-offer/{id}** (importador): recebe `{quantity, clientId}`; chama exportador; compensa em falhas; atualiza ledger.

4. Segurança

- mTLS com CA XporY; JWT de escopo `trader :purchase`; `clock skew ≤ 60s`.

5. Observabilidade

- Traçar hop importador→exportador; métricas de p99, taxa de erro, taxa de idempotência.

6. SLOs

- p99 POST `/trader/purchase` \leq 500ms; erros < 0,5%; 100% de requisições com `Idempotency-Key` válido.

7. Testes

- Contract tests, E2E com falha intermitente, *half-open* CB, pedidos duplicados.

8. Playbooks

- Queda do exportador; divergência de estoque; saturação de CB; vazamento de idempotency keys.

9. Rollout

- *Shadow calls* em piloto; *canary* por par; *kill switch* de rota.